Two Nondeterministic Event Building Methods derived from the Barrel Shifter

Gábor Harangozó
Department of Process Control
Budapest University of Technology
H-1521 Budapest, Hungary
E-mail: gabriel@seeger.fsz.bme.hu

Abstract

At High Energy Physics experiments, extensive parallelism allows scalable, high-bandwidth data acquisition systems. On-line event building of physical events is only feasible by using switch-based event builders. The Barrel Shifter is a well-known event building method for switch-based event builders. Two nondeterministic versions of the Barrel Shifter introduced in this paper provide cost-effective alternatives to the Barrel Shifter. Simulation results are presented to show the source buffer requirements of both nondeterministic methods and the Barrel Shifter at different types of detector data flow.

1. Introduction

High Energy Physics (HEP) is a special field of particle physics where charged particles are accelerated and collided. When particles interact in a detector and the interaction satisfies certain conditions, the detector electronics produces a lot of data, which is called an *event*. Events are composed of several event fragments. Each event fragment is produced by a detector segment. The aim of the event building process is to collect the separated, parallel event fragments of the same event in one destination device for off-line processing.

At High Energy Physics experiments, data flow is essentially unidirectional, i.e. from the detector subsystems to a farm of processors or storing devices. Since data volume to be transmitted can reach a few Gigabytes per second [1][2][3], data acquisition systems based on a single shared bus cannot be used any more due to the limited bandwidth. Parallelism is the only solution to eliminate the bandwidth limitation. The use of multiple interconnection results in a scalable, high-throughput event builder, which itself may constitute a small data acquisition system or may be a component of a large, multilayered system. Several implementations of an interconnection network are possible, such as multiple busses, multiport memories or a switching fabric. Both the multiple bus and the multiport memory architectures still have the problem of realization if the number of data sources and data destinations is great. For large event builders, only switched networks based on high-speed serial links seem to be feasible. The most efficient utilization of the switch occurs when several events are simultaneously built, and the event fragments of different events are distributed uniformly in time and space over the switch. This can be achieved by using proper event building methods.

Event building methods may be classified as deterministic and nondeterministic methods. At a deterministic event building method, the event-destination assignment is static, i.e. the destination of all events is predefined. However, at a nondeterministic method, the event-destination assignment is dynamic, i.e. destination of the events is chosen only during the data acquisition run.

This paper is written for data acquisition system designers. First, a switch-based, technology-independent event builder architecture is described that is used to test the event building methods. Afterwards, the token passing mechanism of the Barrel Shifter, the BSM and IBSM methods is introduced. The BSM and the IBSM methods are derived from the Barrel Shifter. Finally, simulation results are presented to show the source buffer requirements of all the three methods at different types of detector data flow.

2. A switch-based event builder architecture

In this chapter, a switch-based, technology-independent event builder architecture is described. This architecture is used in the simulations to test the event building methods. Real data acquisition systems may consist of several layers of such event builders, but even a single event builder may constitute the core of a data acquisition system.

The event builder is based on a symmetric, switched interconnection network, in which the number of sources and destinations is equal, as shown in Figure 1. The behavioral model of the architecture contains a few simplifying conditions in order to obtain simulation statistics and to concentrate strictly on the event building process. Description of the architecture and its behavior is summarized hereinafter.

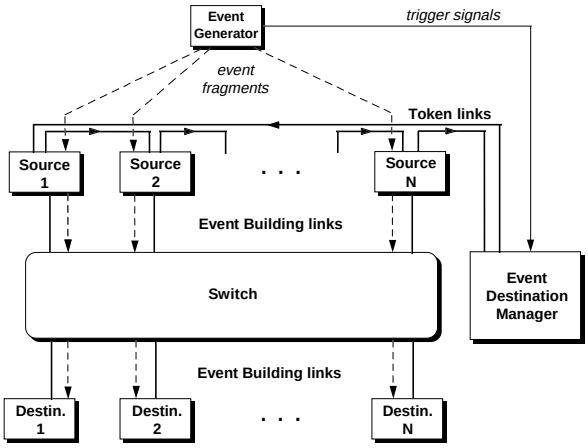

Figure 1. Architecture of a switch-based event builder

The *Event Generator* creates event fragments and trigger signals during the simulations, thus it may be regarded as either the detector front-end electronics or the output of a preceding event building level. If both the size and the trigger interarrival time of the event fragments are constant, the detector data flow is *deterministic*, otherwise it is regarded as *stochastic* data flow. When the size of the event fragments is not constant (i.e. their size follows random distribution), the event fragments of one event are assumed to be independent. The event fragments are sent to the s*ources*, whereas the trigger signals are sent to the *Event Destination Manager* (EDM) within zero time. Sources and *destinations* are connected to the *Switch* via high-speed serial links referred to as *Event Building links*. In Figure 1, dashed lines with arrow show the direction of the detector data flow. Since event fragments are very large messages in most experiments, the Switch makes circuit-switched connections between the sources and the destinations for each event fragment in order to minimize the overhead of data transfer. Due to the circuit-switched operation mode, there is no need of memory in the Switch. Sources have infinite buffers thus we can examine the maximum source buffer occupancies at the different event building methods. The destinations have buffers only for a maximum size event. After an event is completely assembled, the corresponding destination sends it to a storing device or to the following event building level. During the simulations, events are removed from the system by the destinations within zero time. When an event is removed, its destination is said to be *released*.

Tokens are used to arbitrate the access to the destinations. A token is assigned to an event and it contains the identifier of the destination where the event is to be collected. Tokens are generated by the Event Destination Manager. In an N x N system, at most N tokens are allowed to circulate amongst the sources, all the others are stored in a *Token Queue* of the Event Destination Manager. After a source has sent an event fragment, it passes the token of the event to the next source. After sending the last event fragment of a given event, a source sends the token of the event back to the Event Destination Manager. The Event Destination Manager removes the token and issues a new token from its Token Queue.

Tokens are passed source by source over an additional ring of high-speed serial links referred to as *Token links*. Token links are unidirectional as shown in Figure 1. (Since token traffic can be very intensive at certain event building methods [4], it is better to separate the token traffic from the detector data flow, thus avoiding to overload the Switch.)

3. Event building methods

In this chapter, the token passing mechanism and the event-destination assignment rules of the event building methods are described. In the case of the nondeterministic methods, the event-destination assignment is based on the First-Come-First-Served (FCFS) principle. It means that the identification number of the released destinations is stored in a queue (*Destination ID Queue*) in the Event Destination Manager, and when building of a new event starts, the destination belonging to the first ID of the queue is assigned to the new event. In the case of the deterministic Barrel Shifter method, the event-destination assignment is defined by a formula, but the FCFS policy always gives the same result as the deterministic assignment scheme.

If a source receives more than one token while transmitting an event fragment, tokens are buffered in a queue in the source. The following token passing rules are

based on an event builder with N sources and N destinations.

3.1. The Barrel Shifter method

The Barrel Shifter method was first introduced in [5]. Several studies have been written considering its applicability for physical event building in packet-switched networks [6][7][8].

The Barrel Shifter method has been originally developed for deterministic data flow. Use of tokens in the event building process gives the possibility for the Barrel Shifter to manage stochastic data flow as well. In this case the method still remains deterministic.

The token passing rules of the Barrel Shifter are the following:
1. The token of event i is first sent from the EDM always to the first source.
2. Source p sends each token to source q, where

$$q = p + 1, \quad \text{if } p = 1, 2, 3, \ldots (N-1);$$

$$q = 1, \quad \text{if } p = N.$$

The event-destination assignment rule of the Barrel Shifter is the following:
Event i is assigned to destination r, where

$$r = [\,(i-1) \text{ MOD } N\,] + 1, \qquad i = 1, 2, 3, \ldots$$

This formula results in Round Robin policy for the destinations. If destinations are chosen by the FCFS principle, the event-destination assignment will work in the same way because events are always completed in their generation order, according to the token passing rules.

Figure 2 shows how event fragments are transmitted in a 6 x 6 event builder at 66% link load when deterministic detector data flow is applied. Small bold arrows with a serial number mark the generation of a new event along the time axis. Destinations are identified by capitals from *A* to *F*. Event fragments of the same event are identified by the serial number of the event. According to Figure 2, events are completed in their generation order, even if the trigger interarrival time or the event fragment size is not constant.

The maximum occupancy value of the source buffers is a very important factor from the point of view of the *hardware design* and *cost*. Since sources send the event fragments in the order of arrival, they may use FIFO memory to buffer the event fragments. The size of the FIFO depends on the source's place in the logical order of the sources. The first source has to maintain a FIFO only for one event fragment, which is always transmitted immediately. However, the last source needs a FIFO for $N + 1$ event fragments, if an N x N switch is used. It means that if the logical order of the sources cannot be foreseen or their logical order can change for some reason during the data acquisition period (e.g. there is no interaction in the detector for a long time), each source should be designed to be capable of buffering $N + 1$ event fragments.

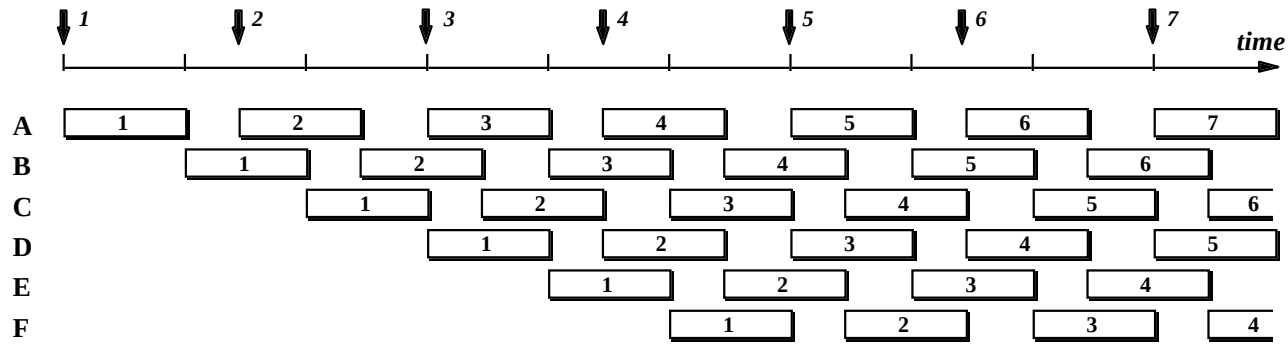

Figure 2. Transmission of event fragments at the Barrel Shifter (at 66% link load)

3.2. The BSM method

The BSM and the IBSM event building methods result evenly distributed buffering amongst the sources, and they decrease the maximum source buffer occupancy of the event builder. The volume of decrease of the maximum source buffer occupancy depends on the load of the Event Building links and on the statistical properties of the detector data flow.

The token passing rules of the BSM method differ depending on whether N is odd or even. For odd N, the rule is the following:

1. The token of event i is first sent from the EDM to source k, where

$$k = [\ (i - 1)\ \text{MOD}\ N\ ] + 1, \qquad i = 1, 2, 3, \ldots$$

2. Source p sends each token to source q, where

$$q = p + 2, \qquad \text{if } p = 1, 2, 3, \ldots (N - 2);$$

$$q = 1, \qquad \text{if } p = (N - 1);$$

$$q = 2, \qquad \text{if } p = N.$$

For even N, the token passing rule is the following:

1. The token of event i is first sent from the EDM to source k, where

$$k = [\ (i - 1)\ \text{MOD}\ N\ ] + 1,$$

$$i = 1, 2, 3, \ldots (N - 1), (N + 1), \ldots$$

$$k = N - 1, \qquad i = N, 2N, 3N, \ldots$$

2. Source p sends each token to source q, where

$$q = p + 2, \qquad \text{if } p = 1, 2, 3, \ldots (N - 2);$$

$$q = 2, \qquad \text{if } p = (N - 1);$$

$$q = 1, \qquad \text{if } p = N.$$

The event-destination assignment is based on the FCFS principle.

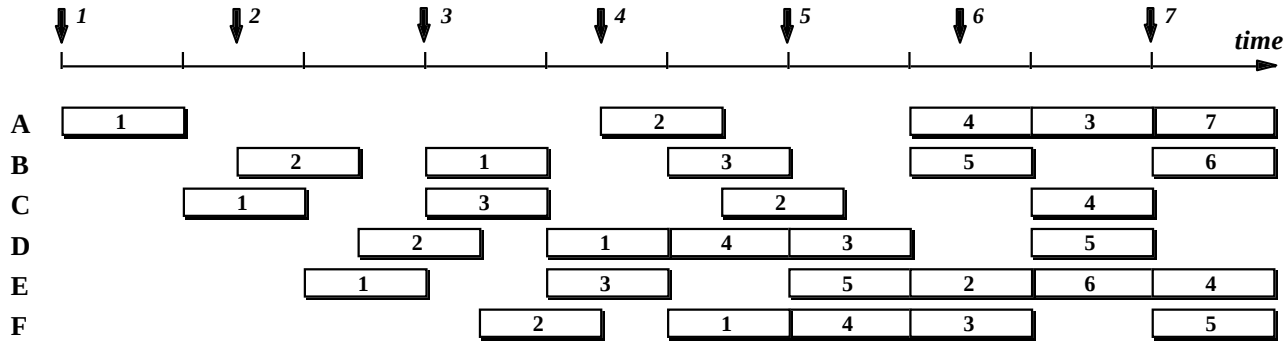

Figure 3. Transmission of equal event fragments at the BSM method (at 66% EB link load)

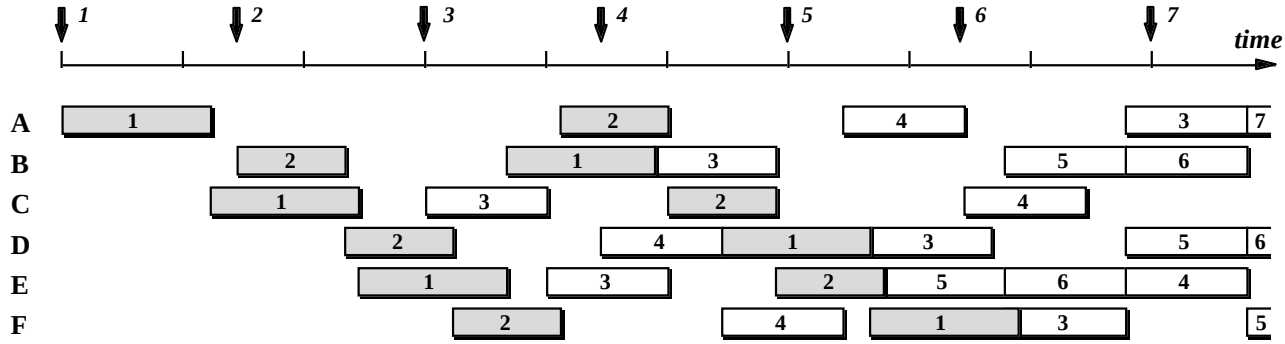

Figure 4. Transmission of different sized event fragments at the BSM method (at 66% EB link load)

Figure 3 shows how event fragments are transmitted in a 6 x 6 event builder at deterministic detector data flow, at 66% Event Building (EB) link load. This figure demonstrates clearly that any of the destinations transmits the event fragments out of their generation order, so instead of FIFO, only random access memory can be used in the sources for buffering.

If event fragments are not equal in length (or the trigger interarrival time is not constant), destinations will be released in varying order. That is why the event-destination assignment is based on the FCFS policy and not on the Round Robin as at the Barrel Shifter. Figure 4 shows an example where *Event 2* has been completed before *Event 1*, due to the different event fragment sizes. (Light gray background of the event fragments emphasizes the different sizes.)

3.3. The IBSM method

The IBSM method is a variant of the BSM method using another token passing mechanism. The token passing rules are the following:

1. The token of event i is first sent from the EDM to source k, where

$$k = [\, 2(i-1) \ \text{MOD} \ N \,] + 1, \qquad i = 1, 2, 3, \ldots$$

2. Source p sends each token to source q, where

$$q = p + 1, \qquad \text{if } p = 1, 2, 3, \ldots (N-1);$$

$$q = 1, \qquad \text{if } p = N.$$

The event-destination assignment is based on the FCFS principle.

Figure 5 shows how event fragments are transmitted at 66% EB link load and at deterministic detector data flow. At this method – as well as at the BSM method – only random access memory can be used in the sources for buffering since event fragments are not sent in their generation order.

Figure 6 shows an example where different size of the event fragments causes the events to be completed out of their generation order. In Figure 6, *Event 2* has been completed before *Event 1*.

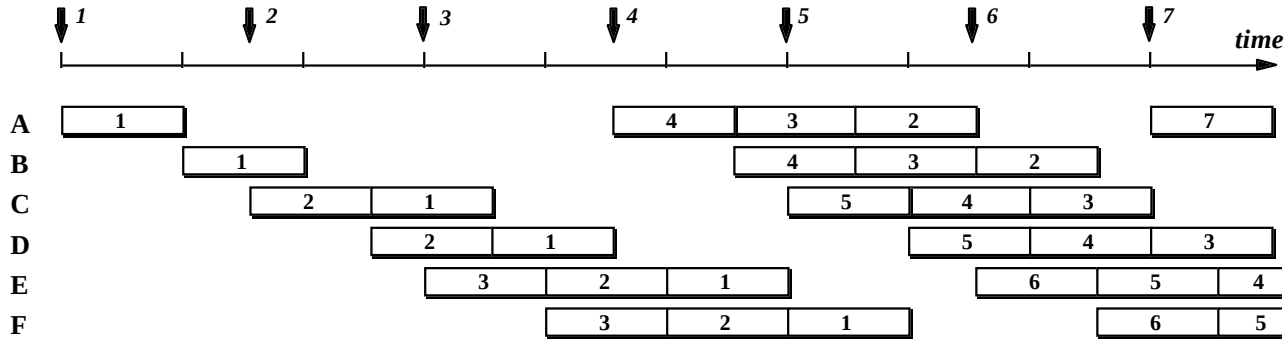

Figure 5. Transmission of event fragments at the IBSM method (at 66% EB link load)

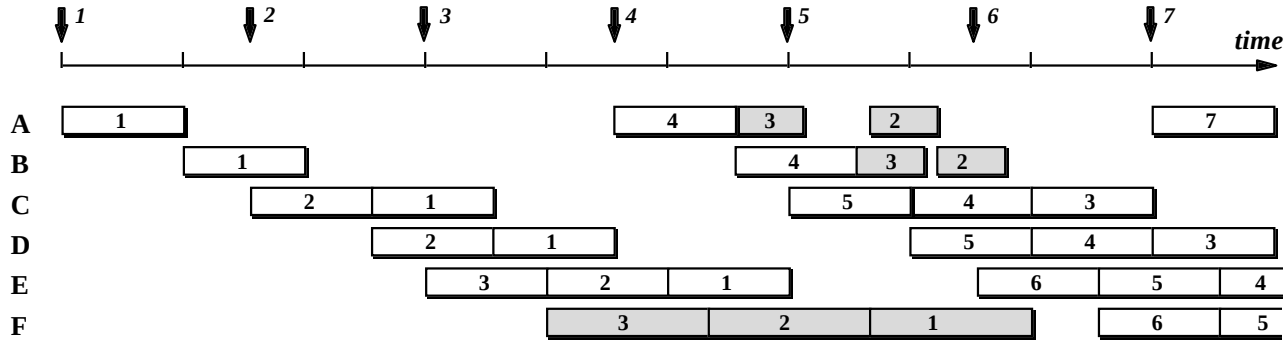

Figure 6. Transmission of different sized event fragments at the IBSM method (at 66% EB link load)

4. Performance analysis by simulation

Formal performance analysis of an event building system with stochastic detector data flow is not available because sources may transmit event fragments out of their arrival order. Therefore computer simulation seems to be the only tool to get statistics on the parameters of interest. Simulations have been completed to compare the Barrel Shifter (BS), the BSM and the IBSM methods at deterministic and different types of stochastic data flow.

The aim of the simulations is to show the *maximum source buffer occupancy* as a function of the trigger rate. This parameter is an observed value of one simulation run and it gives the maximum buffer occupancy that is detected among the sources.

The simulator is written in MODSIM II, which is a general purpose, object-oriented language supporting discrete-event simulation [9]. A simple 8 x 8 circuit-switched event builder is simulated. The mean event fragment size is 1 kbyte. The Event Building link bandwidth is set to 1 Mbyte/s. The transmission time of the event fragments is proportional to their size. There is no dead time on the Event Building links, neither in the Switch. The token passing time is assumed to be negligible as compared to the transmission time of the event fragments and therefore it is set to zero. Simulation statistics are based on steady-state observations. The total number of observed events is 10000, including events of the initial transient.

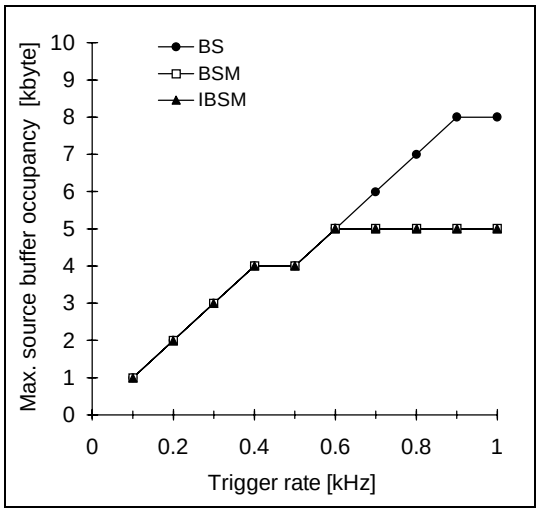

Figure 7. Maximum source buffer occupancies at deterministic data flow

Figure 7 shows the maximum source buffer occupancies at deterministic data flow. The values of this figure can also be obtained by calculations. If the trigger rate is greater than 900 Hz, i.e. the EB link load is higher than 90%, the smaller maximum source buffer occupancy will be almost 40% less at the BSM and the IBSM methods than at the Barrel Shifter. For larger event builders this reduction ratio is even less and converges to 50% as the number of sources increases. If the trigger rate is less than 500 Hz, all the three event building methods have the same performance regarding the maximum source buffer occupancy.

In Figure 8, the maximum source buffer occupancies are shown when the event fragment size is constant and the trigger interarrival time follows Exponential distribution. The nondeterministic methods perform about 20-30% smaller maximum source buffer occupancy than the Barrel Shifter. Above 800 Hz (80% EB link load) the maximum occupancies start to increase dramatically (not shown in the figure) and go to infinity as the trigger rate converges to 1kHz.

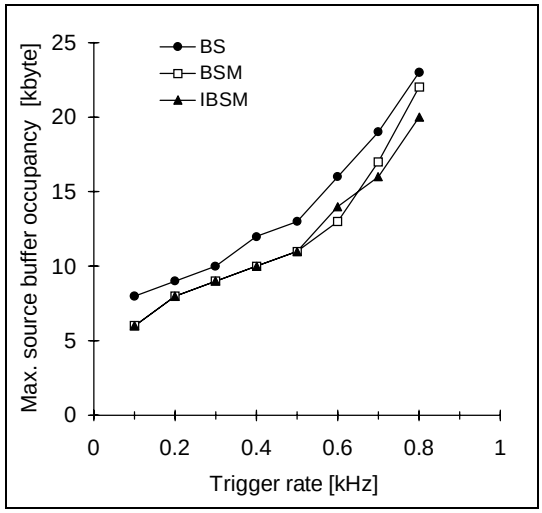

Figure 8. Maximum source buffer occupancies at Exponential trigger interarrival time and constant event fragment size

Figure 9 shows the maximum source buffer occupancies when the trigger interarrival time follows Exponential distribution and the event fragment sizes follow Gaussian distribution with a relative standard deviation of 20% of the mean value. At medium, 30-60% EB link load, the nondeterministic methods result in about 20-40% smaller maximum buffer occupancy than the Barrel Shifter, otherwise all the three methods give similar results. Above 600 Hz (60% EB link load), the maximum occupancy values increase steeply and go to infinity as in the previous case.

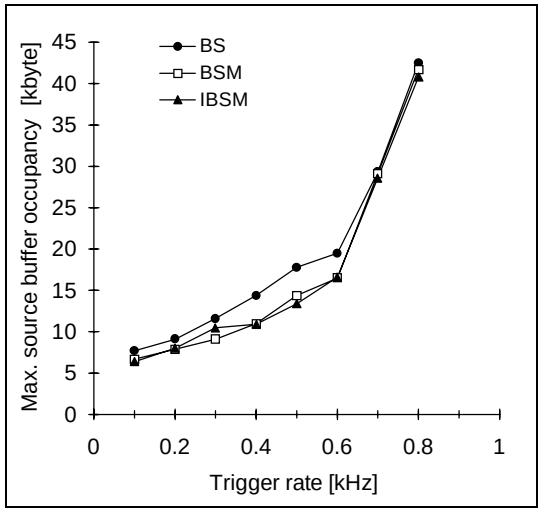

Figure 9. Maximum source buffer occupancies at Exponential trigger interarrival time and Gaussian event fragment sizes with 20% relative std. deviation

Although the event fragments are distributed near evenly amongst the sources when using a nondeterministic method, the whole system buffers more event fragments in average than at the Barrel Shifter. This results from that the event fragments spend more time in the buffers at the BSM and the IBSM methods in average, due to the token passing rules. One consequence of the increased buffering time of the event fragments is that the *average event building latency* will be longer at the BSM and the IBSM methods. The event building latency is defined as the time needed to collect all event fragments of a given event after its trigger signal.

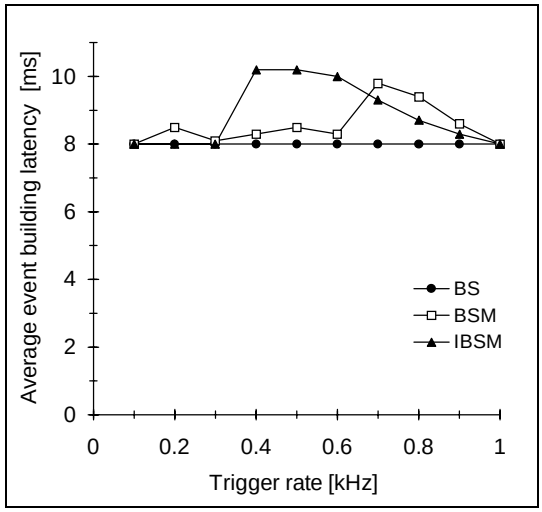

Figure 10. Average event building latencies at deterministic data flow

Regarding the average event building latencies, significant difference between the three methods arises only at deterministic data flow. Figure 10 shows the average event building latencies as a function of the trigger rate. At 1 kHz (100% EB link load), all the three methods give the same latency. However, if the trigger rate is less than 1 kHz, the BSM and the IBSM methods result in up to 20-30% longer average event building latency than the Barrel Shifter.

5. Summary and conclusions

In data acquisition systems of High Energy Physics experiments, interconnection networks are used to transfer detector data to the storing devices. Switched networks provide high-performance, scalable event building systems. In order to best utilize the aggregate bandwidth of the switch, several event building methods are available, depending on the properties of the data flow. In this paper, two nondeterministic event building methods were introduced and compared to the deterministic Barrel Shifter method.

The well-known Barrel Shifter is easy to implement since FIFO memory can be used for buffering in the sources. However, buffered event fragments are not evenly distributed among the sources which increases the maximum source buffer occupancy of the system.

The BSM and the IBSM methods reduce the maximum source buffer occupancy and provide evenly distributed buffer requirements. Since event fragments are transmitted out of their generation order, only random access memories can be used for buffering in the sources.

Simulation results were presented to compare the maximum source buffer occupancies at different kinds of data flow. If the event fragment size and the trigger rate are constant and the link load is high, the BSM and the IBSM methods may result near 50% smaller maximum source buffer occupancy as compared to the Barrel Shifter. In any other cases, the two nondeterministic methods also result some decrease in the maximum source buffer occupancy, but the reduction is not so significant. Although the maximum source buffer occupancies are less at the BSM and the IBSM methods due to the evenly distributed buffering, the whole system buffers more event fragments in average than at the Barrel Shifter. Therefore the average event building latencies are generally longer at the BSM and the IBSM methods.

Examination of the effects of packet-switched operation and finite source buffers is the task of future. Implementation of technology specific architectures is also included in the future investigations.

References

[1] E. Barsotti, A. Booth, M. Bowden: "Effects of various event building techniques on data acquisition system architectures", Fermilab Internal note, Batavia, USA, April 1990.

[2] ALICE Collaboration: "Technical Proposal for a Large Ion Collider Experiment at the CERN LHC", CERN Internal note, LHCC 95-71, Geneva, Switzerland, December 1995.

[3] CMS Collaboration: "Technical Proposal", CERN Internal note, LHCC 94-38, Geneva, Switzerland, June 1994.

[4] G. Harangozo: "Simulation of Event Building Methods in High Energy Physics Data Acquisition Systems", Proceedings of the European Simulation Multiconference 1996, Budapest, Hungary, 2-6 June, 1996.

[5] M. Bowden, H. Gonzales, S. Hansen, A. Baumbaugh: "A High-Throughput Data Acquisition Architecture Based on Serial Interconnects", IEEE Transactions on Nuclear Science, Vol.36, No.1, p. 760-764, February 1989.

[6] M. Letheren, J. Christiansen, I. Mandjavidze: "An Asynchronous Data-Driven Event Building Method Based on ATM Switching Fabrics", CERN Internal Note, ECP 93-14, Geneva, Switzerland, November 1993.

[7] T. Lazraq, H. Tenhunen: "Performance Evaluation of An Event Builder Based on An ATM Switching Fabric with An Internal Link-Level Hardware Flow-Control Protocol", CERN Internal note, ECP 93-24, Geneva, Switzerland, December 1993.

[8] J. Christiansen, J-P. Dufey, M. Letheren, I. Mandjavidze: "NEBULAS: A high performance data-driven event building architecture based on an asynchronous self-routing packet-switching network", CERN Technical Report, DRDC 93-55, Geneva, Switzerland, December 1993.

[9] CACI Product Company: "MODSIM II, The Language for Object-Oriented Programming", Reference Manual, CACI Products Company, La Jolla, USA, May 1991.